Multiplication-Free Biometric Recognition for Faster Processing under Encryption

Amina Bassit [1], Florian Hahn [1], Raymond Veldhuis [1,3], Andreas Peter [1,2]
[1] University of Twente, Enschede, The Netherlands, [2] University of Oldenburg, Oldenburg, Germany
[3] Norwegian University of Science and Technology, Gjøvik, Norway
{a.bassit, f.w.hahn, r.n.j.veldhuis}@utwente.nl, andreas.peter@uol.de

2022 IEEE International Joint Conference on Biometrics (IJCB) | 978-1-6654-6394-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCB54206.2022.10007958

Abstract

The cutting-edge biometric recognition systems extract distinctive feature vectors of biometric samples using deep neural networks to measure the amount of (dis-)similarity between two biometric samples. Studies have shown that personal information (e.g., health condition, ethnicity, etc.) can be inferred, and biometric samples can be reconstructed from those feature vectors, making their protection an urgent necessity. State-of-the-art biometrics protection solutions are based on homomorphic encryption (HE) to perform recognition over encrypted feature vectors, hiding the features and their processing while releasing the outcome only. However, this comes at the cost of those solutions' efficiency due to the inefficiency of HE-based solutions with a large number of multiplications; for (dis-)similarity measures, this number is proportional to the vector's dimension. In this paper, we tackle the HE performance bottleneck by freeing the two common (dis-)similarity measures, the cosine similarity and the squared Euclidean distance, from multiplications. Assuming normalized feature vectors, our approach pre-computes and organizes those (dis-)similarity measures into lookup tables. This transforms their computation into simple table-lookups and summation only. We study quantization parameters for the values in the lookup tables and evaluate performances on both synthetic and facial feature vectors for which we achieve a recognition performance identical to the non-tabularized baseline systems. We then assess their efficiency under HE and record runtimes between 28.95*ms and* 59.35*ms for the three security levels, demonstrating their enhanced speed.*

1. Introduction

Modern biometric recognition technologies lean on deep neural networks (DNNs) to extract distinctive representations of biometric samples (i.e., facial images), called feature vectors. Thus, the biometric recognition task becomes the comparison of two feature vectors against each other by calculating a (dis-)similarity score, usually via the cosine similarity or the squared Euclidean distance (SED). Studies have shown that from those feature vectors, it is possible to infer personal information (e.g., gender, age, health condition, ethnicity, occupation, etc.) [1] and reconstruct raw biometric samples (e.g., facial images) of individuals, known as model inversion attacks [2]. The plaintext access to those feature vectors enables, on the one hand, an undesirable personal information inference that intensifies the severeness of social issues, such as gender inequality and discrimination, due to biased decision-making models. Such models violate the privacy of individuals by performing classification tasks other than recognition. On the other hand, it permits the reconstruction of the raw biometric sample from a given reference template or probe, which leads to security issues such as identity fraud and impersonation attacks. Therefore, feature vectors extracted from DNNs are extremely sensitive and require strong protection.

Biometric template protection schemes (BTPs) [3] try to preserve biometric information (e.g., biometric feature vector) with a maintained recognition performance. BTPs come in different flavors, each approaching the biometrics privacy challenges with distinct techniques (e.g., helper data and Bloom filters). Among existing BTPs, homomorphic encryption (HE) based BTPs [4–8] seem promising in tackling these issues since they carry both the biometric data and its processing to an encrypted domain. Generally, HE-based BTPs compare two encrypted biometric feature vectors against each other by measuring a (dis-)similarity score under encryption and then delivering, to the party of interest, either 1) a final score (followed by a clear-text comparison with the threshold [4–7] yielding the recognition decision) or 2) the final recognition decision (that was preceded by an encrypted comparison with the threshold [8]). Calculating such a (dis-)similarity score under encryption involves a number of homomorphic multiplications proportional to the feature vector's dimension. The failure of HE to handle computations with many multiplications hurts those solutions' efficiency. For instance, to calculate the SED under encryption using the CKKS encryption scheme [9], [7] takes 2.11s for 128-dimensional encrypted feature vectors while for 512-dimensional vectors [5] runs in 5s and [6] runs in 3.39s for 128bits security level; using another HE scheme, BFV [10], the runtime improves to 0.85s [5] and 0.61s [6] but still improvable.

To reduce the number of homomorphic multiplications, the

state-of-the-art [4] adopts the *single-instruction multiple-data* (SIMD) and the plaintext packing properties of fully HE to decrease this number to one homomorphic multiplication for calculating the inner product (IP) under HE over encrypted normalized feature vectors, which corresponds to the cosine similarity. In [4], the author considers only the clear-text comparison with the threshold setting, which has the downside of exposing the final score. The knowledge of the final score can lead to the reconstruction of the original biometric template, as shown by [11–13]. In contrast to the encrypted comparison setting, the comparison with the threshold performed inside the encrypted domain reveals only one bit of information (*match / no match*). We compare our approach with [4] in Section 6. The first multiplication-free biometric recognition (MFBR) scheme is the homomorphically encrypted log likelihood-ratio classifier (HELR) introduced in [8]. It pre-computes the log likelihood-ratio (LLR) and organizes it into lookup tables, reducing the biometric recognition into three operations: selection of the individual scores from the tables, their addition to calculate a final score, and the comparison of the final score with the biometric threshold. However, to determine a score, this classifier requires prior knowledge about the statistics of the features learned from training the LLR. In general, this prior knowledge is hard to acquire for large-scale applications. Hence, the HELR classifier requires training data, and homomorphic multiplications represent a burden for biometrics, motivating us to tackle these challenges.

In this paper, we present two MFBR schemes for IP and SED. Our solutions are built upon the HELR framework but applied to the IP and SED measures that do not require training. Assuming normalized feature vectors extracted from a well-trained DNN, we determine the probability density function (PDF) and cumulative distribution function (CDF) corresponding to the projection of a point on the unit d-ball upon which we generate the lookup tables (that we call MFIP and MFSED) in an equiprobable manner to reinforce their security. We evaluate the biometric performance of our tables on synthetic and facial feature vectors and achieve a performance identical to their baseline measures, preserving the biometric accuracy. Furthermore, we integrate our MFBRs with HE and compare their runtime against our baseline systems[1] for three security levels. For the clear-text comparison with the threshold, our results show a runtime improved by a factor of 22 (resp. 42 and 20) for a 128 bits (resp. 192 and 256 bits) security level compared to our baseline systems, achieving similar efficiency as [4]. For the encrypted comparison with the threshold [8], our solutions outperform both the baseline systems and the state-of-the-art [4][2]. Applying the comparison method described in [8] to [4] gives runtimes 10 times slower than those of the clear-text comparison due to its quantization approach that yields a very large score range; while our solution results in runtimes that are only 3 times slower than in the clear-text comparison.

In summary, we make the following contributions:

- We propose two MFBRs implementing IP and SED comparison measures that do not require training.
- We experimentally investigate the MFIP and MESED tables' parameters, evaluate their biometric performance, and achieve a performance identical to IP and SED.
- We integrate our MFBRs with HE, evaluate their runtime efficiency, and record a runtime faster than our baseline systems but similar to [4] in the clear-text comparison while outperforming both in the encrypted comparison with the threshold.

2. Background: HELR Framework

The HELR classifier assumes that the features are independent and follow the Gaussian distribution. The features' independency allows treating each feature separately and thus calculating the LLR per feature. Therefore, in the following, we describe the HELR framework process for a given feature; the same applies to all features.

2.1. Generation of HELR lookup tables

For a single feature, the LLR is a two-input-one-output function. Pre-computing it yields a lookup table where the rows' (resp. columns') indexes represent the possible values of the first (resp. second) input, feature from the reference template (resp. probe) and the cells contain the output, individual scores. The rows' and columns' indexes result from a feature quantization that maps the continuous domain of real numbers to a finite set of integers, which is needed to limit the possible feature values. In HELR, the feature quantization on $N = 2^n$ *feature quantization levels* is performed by dividing the PDF of the zero-mean and unit variance Gaussian distribution $\mathcal{N}(0,1)$ in an equiprobable manner so that the lookup table's cells have an identical probability for an arbitrary feature observation. Assuming features with Gaussian distribution, they follow $\mathcal{N}(0,1)$ and the bins' borders are determined by following Algorithm 1 where ICDF$(p,0,1)$ is the inverse of the CDF of a $\mathcal{N}(0,1)$ at the cumulative probability p and it returns the value associated with p. In Algorithm 2, $a_i \in \mathrm{Bn}_{a_i}$ (resp. $b_i \in \mathrm{Bn}_{b_i}$) denotes the measured value for feature from the first (resp. second) sample and is quantized to $\hat{a_i}$ (resp. $\hat{b_i}$) using the same Bn bins' borders array of the i-th feature.

2.2. Biometric Recognition based on HELR

Once the lookup tables are generated, the HELR has the following conventions for the reference template and the probe. The d-dimensional feature vector corresponding to the reference template is mapped to its integer representation according to the feature quantization procedure (Algorithm 1 and Algorithm 2).

[1] Our baseline systems use the same quantization as MFBR and comprise two HE multiplications: one for the element-wise multiplication of the coordinates and another one for the isolation of the first plaintext containing the IP.

[2] Using the recent encrypted comparison approach of [8]. If there exists an efficient comparison under encryption algorithm suitable for large score ranges or not dependant on score ranges then [4] would probably perform as fast as ours.

Algorithm 1: Procedure to determine the bins' borders. Algorithm from [8].

Input: $N=2^n$ feature quantization levels
Output: Bn array containing the bins' borders
Bn array of size $N-1$;
for $j \leftarrow 1$ **to** $N-1$ **do**
  $p=j/N$;
  $\text{Bn}[j]=\text{ICDF}(p,0,1)$;
end for

Thus, the HELR reference template becomes a vector of rows selected from the lookup table corresponding to a feature and its index is indicated by the quantized value of that feature. The same feature quantization is applied to the probe feature vector but a feature quantized value indicates the column's index of the row corresponding to that feature in the reference template. Hence, given the HELR reference template and an HELR probe, the biometric recognition reduces to the row-wise selection of the individual scores from the reference template based on the probe. Then, their addition produces a final score S which is compared against a biometric threshold Θ. The case where S exceeds Θ is considered a *match*, otherwise it is a *non-match*.

Algorithm 2: Feature quantization on $N=2^n$ feature levels. Algorithm from [8].

Input: a_i raw feature value of the i-th feature and
  Bn array containing the bins' borders of the i-th feature
Output: $\hat{a}_i$ quantized value
for $j \leftarrow 1$ **to** $N-1$ **do**
  if $a_i < Bn[j]$ **then return** $j-1$;
end for
return $N-1$

3. Multiplication-Free Biometric Recognition

Our primary goal is to apply the HELR framework to common (dis-)similarity measures not requiring training to learn the features' statistics for determining a score, such as the cosine similarity and the squared Euclidean distance. To construct suitable lookup tables, we need to determine 1) the table's cell borders by equiprobably dividing the proper PDF that represents a random observation of the features, 2) the proper PDF and its CDF, and 3) the representative value of a cell. For this purpose, finding the proper PDF and its CDF that define the feature vectors resulted from a DNN might be tricky and dependent on the DNN's training and architecture, which may lead to a different distribution per architecture. To avoid this, given that the feature vectors resulting from a DNN are points spread on the $\mathbb{R}^d$ space, we normalize them to bring them on the surface of the unit d-ball. Note that this normalization does not affect the (dis-)similarity measures on non-normalized feature vectors. A normalized vector of dimension d can be seen as a point on the unit d-ball. We assume that the points on the unit d-ball are uniformly distributed. This assumption is justified by the fact that a well-trained DNN (i.e., feature extractor) can achieve an (near-)optimal recognition performance only if the features are uniformly distributed over the d-ball. Thus, we derive the PDF of the projection of a point on the d-ball on an axis to determine the PDF corresponding to the coordinates of the normalized vector. Based on the inverse CDF of this PDF, we equiprobably quantize the observed projected point following Algorithm 1 and Algorithm 2 using the inverse CDF of the point projection on the unit d-ball instead of the ICDF$(p,0,1)$.

Remark 3.1 ((Dis-)Similarity Measure Equivalent to the Inner Product)**.** *Note that the cosine of normalized vectors equals to their inner product. While the squared Euclidean distance of two normalized vectors is equivalent to their inner product via the monotonic function* $x \rightarrow 2(1-x)$.

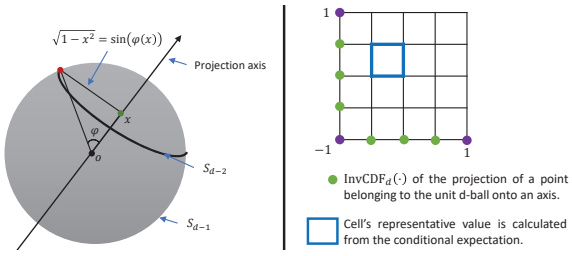


Figure 1: Illustration of the projection of a point belonging to the surface area of the unit d-ball onto the x-axis (left) and an example of a MFBR table with $2^2=4$ feature quantization levels and feature vectors of dimension d (right).

3.1. Point Projection on the Unit d-ball

In this section, we describe how we derive the PDF and its CDF corresponding to the projection of a point on the unit d-ball. Note that this PDF is the PDF of every coordinate of a d-dimensional normalized feature vector.

3.1.1 PDF of the Point Projection on the Unit d-ball

Let X denote the random variable after projection on the x-axis. Let x denote a realization of that random variable. Let $\varphi(x)$ denote the angle of a point on the d-ball that is projected onto x. Let S_{d-1} denote the surface of a d-ball given by $\text{S}_{d-1}(r)=\frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})}r^{d-1}$ where r denotes the radius and Γ denotes the Gamma function $\Gamma(z)=\int_0^\infty e^{-t}t^{z-1}dt$. The probability mass of the points on the d-ball is uniformly distributed over the d-ball. The probability mass that a point on the d-ball is projected onto the x-axis is the mass contained on the surface of the (d-1)-ball with center x and radius $\sqrt{1-x^2}=\sin(\varphi(x))$. In order to derive the PDF $f_X(x)$ we have $f_X(x)=\frac{d}{dx}\Pr\{X\leq x\}$. By looking at Figure 1 we can see that $\Pr\{X \leq x\} = \frac{1}{S_{d-1}(1)}\int_{\varphi(x)}^{\pi} S_{d-2}(\sin(t))\,dt$ where $\frac{1}{S_{d-1}(1)}$ is normalization factor. Hence, $f_X(x) = \frac{-1}{S_{d-1}(1)}S_{d-2}(\sqrt{1-x^2})\varphi'(x)$. Note

that $x=\cos(\varphi) \Longrightarrow \varphi'(x)=\frac{-1}{\sqrt{1-x^2}}$, thus we have

$$f_X(x)=\frac{1}{S_{d-1}(1)}S_{d-2}(\sqrt{1-x})\frac{1}{\sqrt{1-x^2}}$$
$$=\frac{1}{B(\frac{1}{2},\frac{d-1}{2})}\left(\sqrt{1-x^2}\right)^{(d-3)}$$

where the Beta function: $B(x,y)=\int_0^1 t^{x-1}(1-t)^{y-1}dt$. Recall the relation between the Gamma function and the Beta function is given by $B(u,v)=\frac{\Gamma(u)\cdot\Gamma(v)}{\Gamma(u+v)}$. Therefore, the PDF of the projection of a point belonging to the unit d-ball onto an axis is given by

$$f_X(x)=C\cdot\left(\sqrt{1-x^2}\right)^{(d-3)} \tag{1}$$

where $x\in[-1,1]$ and the normalizing constant $C=\frac{1}{B(\frac{1}{2},\frac{d-1}{2})}$.

3.1.2 CDF of the Point Projection on the Unit d-ball

Let F_X denote the CDF corresponding to the PDF f_X. To calculate F_X, we need to solve the integral in Equation (2).

$$F_X(x)=\int_{-1}^{x} C\cdot\left(\sqrt{1-t^2}\right)^{(d-3)}dt \tag{2}$$

where $x\in[-1,1]$. Let us solve the following integral separately $\int\left(\sqrt{1-t^2}\right)^{(d-3)}dt$. Using the substitution $t=\sin(u)$, we have $dt=\cos(u)du$.

$$\int\left(\sqrt{1-t^2}\right)^{(d-3)}dt=\int(\cos(u))^{(d-2)}\,du \tag{3}$$

In Equation (3), the passage from the left-hand side to the right-hand side is justified by the fact that $t=\sin(u)$ implies $u=\arcsin(t)\in[\frac{-\pi}{2},\frac{\pi}{2}]$. Thus, $\cos(u)\in[0,1]$ when $u\in[\frac{-\pi}{2},\frac{\pi}{2}]$. Solving this integral depends on the parity of d. Since d in our case represents the size of a feature vector that is generally expressed as $2^{\#Bits}$, let us assume that d is even which implies that $d-2$ is also even. By using the trigonometric power formula for cosine raised to an even power [14], for $n=d-2$ we have:

$$\cos^n(u)=\frac{1}{2^n}\binom{n}{\frac{n}{2}}+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\cos((n-2k)u) \tag{4}$$

Plugging in (4) in the integral (3), the integral becomes:

$$\int(\cos(u))^n\,du=\int\frac{1}{2^n}\binom{n}{\frac{n}{2}}+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\cos((n-2k)\,u)\,du \tag{5}$$

$$=\frac{1}{2^n}\binom{n}{\frac{n}{2}}\,u+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\frac{\sin((n-2k)\,u)}{(n-2k)} \tag{6}$$

$$=\frac{1}{2^n}\binom{n}{\frac{n}{2}}\arcsin(t)+\frac{2}{2^n}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\frac{\sin((n-2k)\arcsin(t))}{(n-2k)} \tag{7}$$

From (6) to (7), we go back to the original variable by replacing $u=\arcsin(t)$. The term $\sin((n-2k)\arcsin(t))$ in Equation (7) can be simplified using the multiple-angle formula in [14]. That is Equation (8)

$$\sin(rx)=\sum_{j\text{ odd}}^{r}(-1)^{\frac{j-1}{2}}\binom{r}{j}\cos^{r-j}(x)\sin^j(x) \tag{8}$$

Replacing $r=n-2k$, $x=\arcsin(t)$, and $\cos(\arcsin(t))=\sqrt{1-t^2}$, and the simplification of Equation (8) solve the integral in Equation (5). Therefore, the CDF of the projection of a point belonging to the unit d-ball onto an axis is given by

$$F_{X_i}(x)=C\cdot\left[\frac{1}{2^n}\binom{n}{\frac{n}{2}}\arcsin(t)+\frac{1}{2^{n-1}}\sum_{k=0}^{\frac{n}{2}-1}\binom{n}{k}\cdot\frac{1}{(n-2k)}\cdot\right. \tag{9}$$

$$\left.\left(\sum_{j\text{ odd}}^{n-2k}(-1)^{\frac{j-1}{2}}\binom{n-2k}{j}\left(\sqrt{1-t^2}\right)^{n-2k-j}\cdot t^j\right)\right]_{-1}^{x}$$

where $n=d-2$. For the inverse CDF, we use Brent's method implemented in SciPy [15] to compute the numerical inverse of the CDF at a specific point.

3.2. Construction of MFIP and MFSED Tables

Unlike the HELR framework, where there is a lookup table for each feature, the pre-computed inner product and the SED require the generation of a single table for all features because they follow the same PDF. Figure (5c) shows the theoretical prediction under the uniformity assumption (the solid line) covers the empirical data (the bar graph), which empirically proves that the points on the unit d-ball are uniformly distributed. Moreover, the classifier's performance can be improved as long as the training samples' feature vectors are not uniform. If the training set is representative of the test data, we may assume that the features are uniformly spread over the ball. We call MFIP the lookup table that pre-computes the inner product and MFSED the one that pre-computes the SED. To generate those tables, we first specify the borders of a cell by equiprobably cutting the x-axis and y-axis according to the PDF of the projection of a point belonging to the unit d-ball onto an axis. Then, we define a table of $N\times N$ cells, where $N=2^n$ denotes the feature quantization levels on n bits. Each cell's borders are determined according to the bins Bn for both axes x and y following Algorithm 1 and Algorithm 2; see Figure 1. Subsequently, we specify the cell's representative value by calculating the joint conditional expectation Equation (10) for independent X and Y. For the MFIP table it is equal to Equation (10) and for the MFSED table it is equal to Equation (11).

$$E_{X,Y}\left[x,y|\text{Bn}\right]=\frac{E_X\left[x|\text{Bn}_x\right]\cdot E_Y\left[y|\text{Bn}_y\right]}{P(\text{Bn})} \tag{10}$$

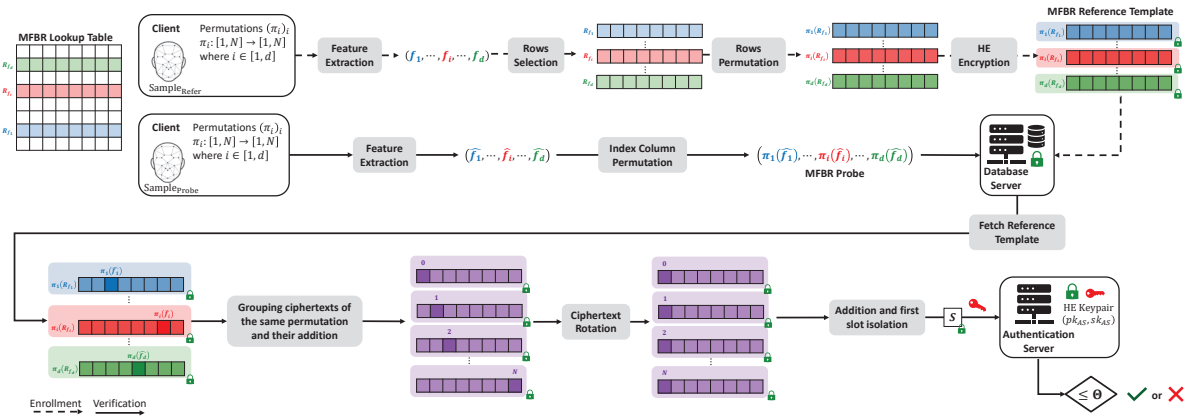


Figure 2: Overview of the integration of MFBR with homomorphic encryption supporting the SIMD and packing properties.

$$E_{X,Y}\left[x,y|\text{Bn}\right] = \frac{\left(E_X\left[x|\text{Bn}_x\right] - E_Y\left[y|\text{Bn}_y\right]\right)^2}{P(\text{Bn})} \tag{11}$$

where Equation (12) defines the expectation over the cell's borders with respect to the x-axis Bn_x (similarly for the y-axis) and Equation (13) represents the probability of a cell. Note that the interval Bn_x has one of the three forms $\left[-1,\text{Bn}_x[1]\right), \left[\text{Bn}_x[j],\text{Bn}_x[j+1]\right)$ or $\left[\text{Bn}_x[n-1],1\right]$.

$$E_X\left[x|\text{Bn}_x\right] = \int_{\text{Bn}_x} x f_X(x)\, dx \tag{12}$$

$$P(\text{Bn}) = \frac{1}{N \times N} \tag{13}$$

Given that the cells' representative values are real-valued, we apply another quantization mapping, that we call *cell quantization*, to render them to integers, making them suitable for homomorphic encryption (HE) schemes with an integer plaintext space. The cell quantization takes the cell's representative value, divides it by a quantization step Δ, and rounds it to the nearest integer.

4. Integration of MFBR with HE

Similarly to the HELR framework, described in Section 2.2, MFBR-based biometric recognition follows the same reference template and probe convention. This convention facilitates the application of a homomorphic encryption layer over the recognition process when both the reference template and probe are protected. The HELR classifier was implemented with an additively homomorphic encryption scheme (additive ElGamal encryption) where the encrypted reference template comprises the rows encrypted component-wise making the number of the ciphertexts, constituting the encrypted reference template, proportional to the sum of the rows' sizes. The integration of MFBR with HE, supporting the SIMD and the plaintext packing properties, compresses this proportionality to the number of rows such as each ciphertext encrypts one row. This is also applicable for HELR classifier when implemented with an HE scheme supporting these two properties.

Figure 2 depicts an overview of the usage of our MFBR scheme for a clear-text comparison with the biometric threshold. For an encrypted comparison with the threshold, the procedure described in Section V-A in [8] is also suitable for our MFBR scheme. The setting of Figure 2 is a semi-honest three-party protocol comprising a client, a database server, and an authentication server, assuming no collusion between both servers. The client is the biometric data owner and possesses d permutations $\pi_i : [1,N] \to [1,N]$ that uses them to row-wise permute the reference template during the enrollment phase and locate the specific individual scores corresponding to a probe during the authentication phase. The database server is the holder of the protected reference templates encrypted under the authentication server's public key pk_{AS}. It selects, under encryption, the individual scores and computes the final score by grouping, adding the encrypted rows to be rotated by the same position[3], and then rotating the resulting ciphertext by j positions. This sum-then-rotate reduces the number of rotations to $d-1$ where d is the length of one row. The addition of the rotated ciphertexts and the isolation of the first plaintext slot, using the optimization described in Remark 4.1, form the encrypted final score. Finally, the authentication server learns the recognition outcome by decrypting the encrypted final score and comparing it against its biometric threshold.

Remark 4.1 (Multiplication-Free First Plaintext Slot Isolation)**.** *Let $v = (0, v_2, \cdots, v_r)$ be a randomly sampled vector of size equals to the ring dimension with a zero in its first coordinate. The first plaintext slot of a ciphertext ct is isolated by adding ct with an encryption of v (additive blinding), assuming that v is freshly sampled. Note that v can be generated and encrypted beforehand but should not be reused.*

[3]Encrypted rows which indexes are in $\text{Group}(j) = \{i | \pi_i(\hat{f}_i) = j, i \in [1,d]\}$, where $j \in [1,N]$, will be rotated by j positions

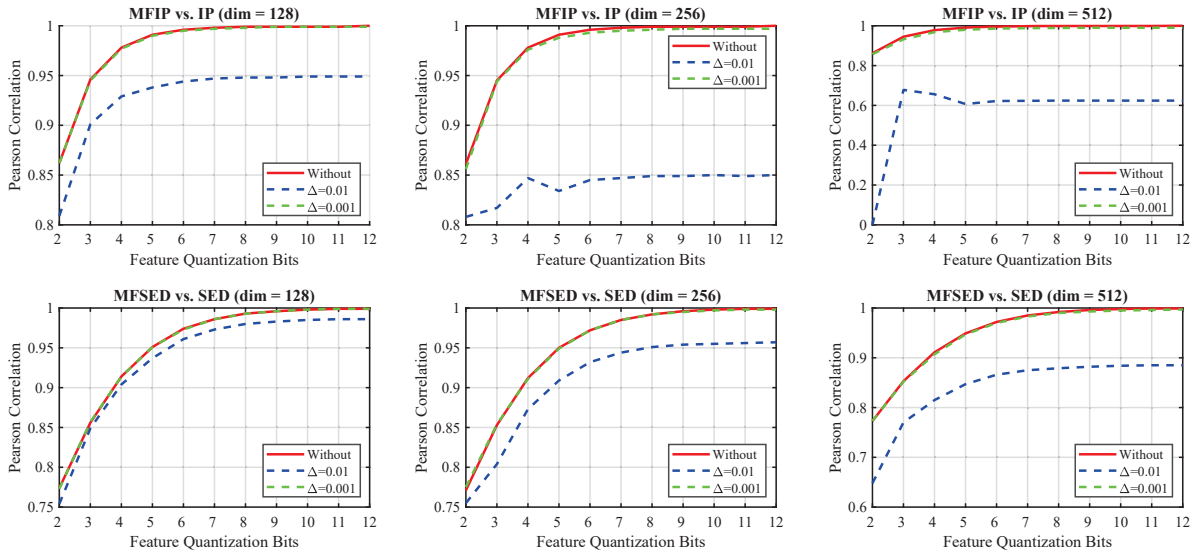


Figure 3: Comparison of MFIP and MFSED against non-quantized and non-precomputed IP and SED expressed in terms of Pearson correlation coefficient for three different dimensions of feature vectors, IP vs. MFIP (first row) and SED vs. MFSED (second row). The x-axis indicates the number of bits (n) used in the feature quantization level. In the solid-line red curve, the lookup table is tested without rounding, while in the dashed-lines blue and green curves, the table is tested with two different rounding values $\Delta = 0.01$ and $\Delta = 0.001$, respectively.

5. Experiments and Evaluations

In this section, we experimentally study the MFIP and MFSED tables in terms of their parameter choices, their biometric performance on a facial dataset, and their runtime performance under encryption. We implement the experiments of Section 5.1 and Section 5.2 in Python 3.9 and the experiments of Section 5.3 in C++ using the PALISADE library [16] for the BFVrns homomorphic encryption scheme and OpenMP [17] for parallelization. We used a Linux Ubuntu 20.04.3 LTS machine run on a 64-bit computer Intel(R) Core i7-10750H CPU with 4 cores (8 logical processors) rated at 2.60 GHz and 16GB of memory. We make our source code publicly available [4].

5.1. Parameters Investigation

The MFIP (resp. MFSED) table is parametrized by a feature vector of dimension d, a feature quantization level 2^n, and a cell quantization step Δ, which we denote as MFIP(d,n,Δ) (resp. MFSED(d,n,Δ)). To understand the performance impact of these parameters, we study the MFIP and MFSED under different combinations of these parameters. We assess the quality of those tables in terms of the Pearson correlation coefficient by generating 200000 synthetic normalized feature vectors. We compare the inner product against the pre-computed inner product from the MFIP and the SED against the pre-computed SED from the MFSED. Figure 3 shows the MFIP and the MFSED tables' quality in terms of Pearson correlation coefficient. We notice that the larger the lookup table's size gets, the more accurate the table gets; the Pearson coefficient converges to 1.

[4] https://github.com/aminabassit/MFBR

The score quantization step Δ has a faint impact on the table's accuracy, starting from values strictly smaller than 0.01, while it has a huge impact on the score range; the smaller Δ gets, the larger the score range becomes. This latter affects the runtime of the encrypted comparison with the threshold setting since it runs fast for smaller score ranges, as mentioned in [8]. The lookup table achieves an optimal accuracy starting from the size $2^3 \times 2^3$ without score quantization and with score quantization for $\Delta = 0.001$ for the three dimensions (128, 256 and 512). For $\Delta = 0.01$, the accuracy is maintained for feature vectors of low dimensions (128), while for high dimensions (256 and 512), the accuracy drops. This is justified by the information loss resulting from the majority of the cells rounded to zero.

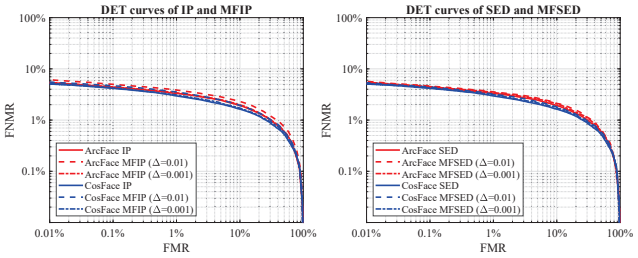


Figure 4: Biometric performance of the baseline systems (IP and SED) and MFBR (MFIP and MFSED) over the VGGFace2 dataset with features extracted from ResNet-100-ArcFace and ResNet-100-CosFace.

5.2. Biometric Evaluation

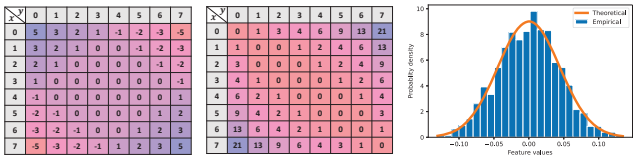
As in [8], our MFBR approach supports any biometric modality that can be encoded as a fixed-length real-valued

feature vector, assuming that the feature extractor is well-trained to yield features uniformly spread over the d-ball. To demonstrate this, we evaluate the biometric performance of the MFIP and MFSED tables on facial feature vectors. We used the VGGFace2 dataset [18] to extract facial feature vectors of dimension 512 using ResNet-100 [19] trained by two different losses: one trained with ArcFace [20] and another one trained with CosFace [21]. In the following experiments, we perform 52500 mated comparisons and 623750 non-mated comparisons.

Figure 4 compares the biometric performance of the baseline IP (resp. SED), as a non-pre-computed and non-quantized approach, against the MFIP (resp. MFSED), as a pre-computed quantized approach, for the following lookup table's parameters: table's size of $2^3 \times 2^3$, $\Delta = 0.01$ and $\Delta = 0.001$. The choice of the first optimal smallest lookup table size is justified by the aim for reference templates of small size since they are formed by a vector of rows belonging to the lookup table. The DETs on Figure 4 show similar performances which is explained by the monotonic relationship between IP and SED (as discussed in Section 3.1), implying an identical relationship between their corresponding pre-computed lookup tables MFIP and MFSED. Although the chosen lookup table's parameters ($2^3 \times 2^3$, $\Delta = 0.01$ and $\Delta = 0.001$) would be expected to yield less performant results compared to the other parameters evaluated in Figure 3, our evaluation results show that they preserve the biometric performance of the baseline non-pre-computed and non-quantized IP and SED measures. As for both score quantization steps, $\Delta = 0.01$ and $\Delta = 0.001$, the DETs are overlapping with the baseline's DETs with a slight performance loss noticed for $\Delta = 0.01$ at the gain of a smaller score range compared to $\Delta = 0.001$ that yields a larges score range. For features extracted by ResNet-100-ArcFace, the baseline IP and SED achieve an EER of 2.76% while their corresponding lookup tables for $\Delta = 0.01$ achieve an EER of 3.14% for MFIP and 3.01% for MFSED and for $\Delta = 0.001$ they achieve an EER of 2.82% for MFIP and 2.88% for MFSED. For features extracted by ResNet-100-CosFace, the baseline IP and SED achieve an EER of 2.47% while their corresponding lookup tables for $\Delta = 0.01$ achieve an EER of 2.85% for MFIP and 2.68% for MFSED and for $\Delta = 0.001$ they achieve an EER of 2.55% for MFIP and 2.53% for MFSED.

5.3. Runtime Evaluation

Our HE-based BTP for the IP baseline system consists of 1) HE multiplying two packed normalized feature vectors that were quantized similarly to MFIP, 2) rotating the resulted ciphertext to the i-th position where $i \in [1,512]$, 3) adding the rotated ciphertexts, 4) HE multiplying the resulted ciphertext by an encryption of $(1,0,\cdots,0)$ to isolate of the first plaintext, and then 5) revealing the final score to compare it against the biometric threshold. Although our IP baseline system uses only two HE multiplications, it has a runtime comparable to previous works [5,6] when using the BFVrns scheme; it runs in 0.67s, 1.21s, and 1.24s for 128, 192, and 256 bits security levels, respectively. We also use it as our SED baseline system since $\text{SED}(u,v) = \text{IP}(u-v,u-v)$. Figure 6 compares the speed of our MFBRs against our baseline systems by measuring their runtime using the BFVrns scheme configured as given in Table 1 [5] over three different security levels (128, 192, and 256bits). For our MFBRs (MFIP and MFSED), we use the best parameters obtained in Section 5.2 (see Figures 5a and 5b) and measure their runtimes according to the description given in Figure 2 using the BFVrns encryption scheme. Although the reference template size of the baseline system (one ciphertext) is smaller than its size in MFBR (512 ciphertexts), our runtime outperforms the baseline system over the three security levels. Figure 6 demonstrates that both MFIP and MFSED are about 22 times (resp. 42 and 20 times) faster than IP and SED for a 128 bits security level (resp. 192 and 256 bits).



(a) MFIP(512,3,0.001) (b) MFSED(512,3,0.001) (c) Theoretical vs. Empirical

Figure 5: MFIP and MFSED tables (Figures (5a) and (5b)) used in Section 5.3. Figures (5c) plots the histogram of a feature coming from a 512-dimensional normalized facial feature vector and its value w.r.t. its projection on the unit d-ball using our derived PDF with $d = 512$.

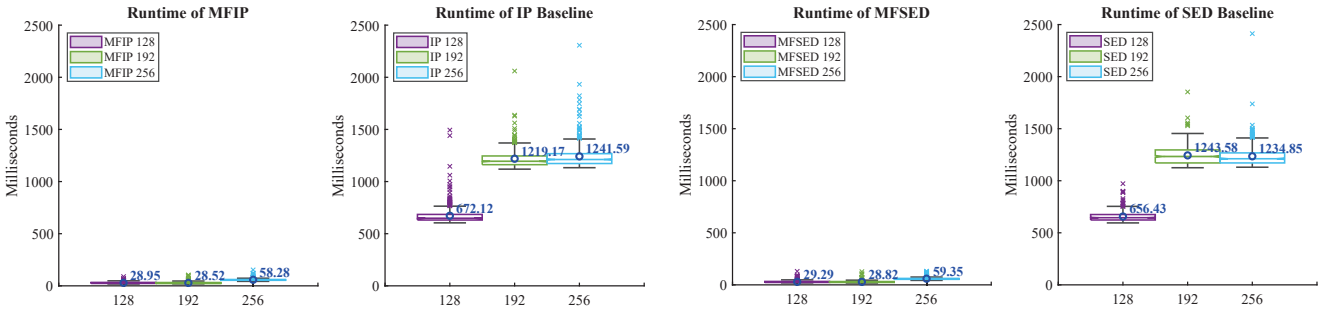
Table 1: PALISADE BFVrns parameters used in MFBR (MFIP and MFSED) and their baseline systems under encryption.

	MFBR			Baseline		
Security level (bits)	128	192	256	128	192	256
Error distribution (σ)	3.2	3.2	3.2	3.2	3.2	3.2
CRT moduli sizes (bits)	36	37	38	36	37	38
Plaintext modulus (p)	65537	65537	65537	65537	65537	65537
$\log_2(p)$	16	16	16	16	16	16
Ciphertext modulus (q)	4.72×10^{21}	1.88×10^{22}	7.55×10^{22}	3.24×10^{32}	2.59×10^{33}	2.07×10^{34}
$\log_2(q)$	72	74	76	108	111	114
Ring dimension (n)	4096	4096	8192	4096	8192	8192

6. Discussions and Future Works

Comparison with the state-of-the-art [4]. The state-of-the-art [4] differs from our baseline by cleverly summing the plaintext slots under encryption, reducing both the number of multiplications to one and the number of rotations to 12. In [4], Algorithm 1 rotates the intermediate ciphertexts by positions of 2^k where $k \in [0,11]$ and accumulates them along the process. This allows the replication of the final score over all the plaintext slots of the final ciphertext, making the first plaintext slot isolation disposable and saving one multiplication over our baseline. For the clear-text comparison with the threshold, a multiplication-free approach and a single-multiplication

[5] These parameters have a complex interdependency and cannot be freely chosen otherwise, the security breaks. The CRT stands for the Chinese remainder theorem, it optimizes the decryption and homomorphic multiplication in the Residue Number System (RNS) [22].



(a) MFIP vs. IP

(b) MFSED vs. SED

Figure 6: Verification runtime of MFBRs (MFIP and MFSED) and our baseline systems (IP and SED) under BFVrns scheme configured following Table 1 and measured over 500 comparisons. The blue circles depict the average runtime.

approach are of similar efficiency; ours meets [4]. Table 2 compares our solution with [4] which we re-implemented in PALISADE library based on its updated code[6]. The advantage of our approach over [4] is visible in the encrypted comparison with the threshold setting. Using the procedure described in Section V-A of [8], our solution is 3 to 5 times faster than [4]. This procedure compares the encrypted final score against all possible scores between the threshold and the maximum score. Thus, it requires a small score range to run efficiently, making it unsuitable for [4] whose feature quantization approach yields a large score range. If there exists an efficient comparison under encryption algorithm supporting large score ranges or independent from them, then [4] would be as fast as ours. Unlike [4], our solution requires more storage space for the template. However, this could be improved by packing all the rows in one ciphertext and enlarging the permutation; we leave this for future work.

Table 2: Comparison between IP baseline system, MFIP, and the state-of-the-art [4] for 512-dimensional feature vectors.

HE-based BTP	Baseline		MFIP		[4]	
Quantization parameters	$n=3$		$n=3, \Delta=0.001$		Precision$=0.0025$	
Mated score range	[5498, 9643]		[−201, 926]		[−33009, 158171]	
# Multiplications	2		0		1	
# Rotations	511		7		12	
# Ciphertexts in reference	1		512		1	
# Ciphertexts in probe	1		0		1	
Security level	128Bits	192Bits	128Bits	192Bits	128Bits	192Bits
Reference template size	199.6KB	396.2KB	68.6MB	68.6MB	199.6KB	396.2KB
Probe template size	199.6KB	396.2KB	2.6KB	2.6KB	199.6KB	396.2KB
Clear-text comparison	672.12ms	1219.17ms	28.95ms	28.52ms	34.66ms	35.79ms
Encrypted comparison	1922.54ms	1936.13ms	67.85ms	123.92ms	337.23ms	340.15ms

Security of our MFBR lookup table. Our lookup table is generated independently from a dataset and does not contain any personally identifiable information. Its cells are generated in an equiprobable manner so that they have an identical probability for an arbitrary feature observation, reinforcing the table's security. We assume that our MFBR lookup table is public knowledge. Because we encrypt the rows using a probabilistic encryption scheme, even encrypting the same row twice results in two completely different ciphertexts. As a result, the encrypted rows cannot be linked to the clear-text rows, whose index provides the quantized feature value. The client can make the server blindly select the specific columns without learning their real values by changing the probe's encoding to the encryption of a vector of ones in the to-be-selected coordinates and zeros elsewhere and multiplying the encrypted probe with the encrypted reference template. However, this introduces one multiplication that we avoid by making the client apply a secret permutation of rows before the reference template encryption and send the indexes' permutation for the selection, enabling a cheap blind selection on the server-side. Future research could further improve the blind selection by exploring the disentanglement of plaintext slots from a packed ciphertext without multiplication and rotation.

[6]Updated code w.r.t the standardized security parameters is at [23]

7. Conclusion

In this paper, we demonstrated that the two common biometric comparison measures (cosine similarity and squared Euclidean distance) can be pre-computed and quantized without biometric accuracy loss. Upon our finding, we succeeded in freeing these comparison measures from multiplications for a smooth application of an encryption layer. The results of our experiments show that our approaches preserve the biometric performance of the pre-computed version of these comparison measures when tested on facial features and improve their runtime under encryption by a factor of 20 to 42. This makes our multiplication-free solution as accurate as the baseline but faster under encryption for both the clear-text and the encrypted comparison with the threshold settings.

Acknowledgement

This work was supported by the PriMa project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 860315.

References

[1] A. Acien, A. Morales, R. Vera-Rodriguez, I. Bartolome, and J. Fierrez, "Measuring the gender and ethnicity bias in deep models for face recognition," in *Iberoamerican Congress on Pattern Recognition*, pp. 584–593, Springer, 2018. 1

[2] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 253–261, 2020. 1

[3] M. Sandhya and M. V. Prasad, "Biometric template protection: A systematic literature review of approaches and modalities," *Biometric Security and Privacy*, 2017. 1

[4] V. N. Boddeti, "Secure face matching using fully homomorphic encryption," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–10, IEEE, 2018. 1, 2, 7, 8

[5] P. Drozdowski, N. Buchmann, C. Rathgeb, M. Margraf, and C. Busch, "On the application of homomorphic encryption to face identification," in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, IEEE, 2019. 1, 7

[6] J. Kolberg, P. Drozdowski, M. Gomez-Barrero, C. Rathgeb, and C. Busch, "Efficiency analysis of post-quantum-secure face template protection schemes based on homomorphic encryption," in *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–4, IEEE, 2020. 1, 7

[7] T. Yang, Y. Zhang, J. Sun, and X. Wang, "Privacy enhanced cloud-based facial recognition," *Neural Processing Letters*, pp. 1–9, 2021. 1

[8] A. Bassit, F. Hahn, J. Peeters, T. Kevenaar, R. Veldhuis, and A. Peter, "Fast and accurate likelihood ratio-based biometric verification secure against malicious adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, 2021. 1, 2, 3, 5, 6, 8

[9] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 409–437, Springer, 2017. 1

[10] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *Annual Cryptology Conference*, pp. 868–886, Springer, 2012. 1

[11] P. Mohanty, S. Sarkar, and R. Kasturi, "Privacy & security issues related to match scores," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pp. 162–162, IEEE, 2006. 2

[12] P. Mohanty, S. Sarkar, and R. Kasturi, "From scores to face templates: A model-based approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 12, pp. 2065–2078, 2007. 2

[13] P. Mohanty, S. Sarkar, and R. Kasturi, "Reconstruction of biometric image templates using match scores," Apr. 24 2012. US Patent 8,165,352. 2

[14] I. Gradshteyn, I. Ryzhik, and R. H. Romer, "Tables of integrals, series, and products," 1988. 4

[15] "SciPy (release 1.9.0)." https://scipy.org/. 4

[16] "PALISADE Lattice Cryptography Library (release 1.11.5)." https://palisade-crypto.org/, 2021. 6

[17] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE computational science and engineering*, vol. 5, no. 1, 1998. 6

[18] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pp. 67–74, IEEE, 2018. 7

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 7

[20] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019. 7

[21] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5265–5274, 2018. 7

[22] S. Halevi, Y. Polyakov, and V. Shoup, "An improved RNS variant of the BFV homomorphic encryption scheme," in *Cryptographers' Track at the RSA Conference*, pp. 83–105, Springer, 2019. 7

[23] V. N. Boddeti, "Source code of: Secure Face Matching Using Fully Homomorphic Encryption." https://github.com/human-analysis/secure-face-matching. 8